

树图编辑工具 (TreeEditor) 使用说明书

詹卫东
北京大学中文系

目 录

一 程序文件与界面	2
1.1 程序文件	2
1.2 程序界面	2
二 树库文件的格式	3
2.1 输入	3
2.2 输出	3
2.2.1 表 (list) 输出格式	3
2.2.2 XML 输出格式	4
2.2.3 导出原始句子文件	4
三 句子结构树的显示与编辑	4
3.1 显示	4
3.1.1 树图的缩小与放大	5
3.1.2 句子列表显示模式切换	5
3.2 编辑	5
3.2.1 节点拖动	5
3.2.2 节点删除	5
3.2.3 修改标签	6
3.2.4 插入节点	6
3.2.5 恢复	6
3.2.6 整句删除	6
3.2.7 字符串编辑模式	6
四 辅助编辑功能	7
4.1 跳转	7
4.2 查找	7
4.3 续查	8
4.4 高级查找	8
4.5 子树替换	13
4.6 查错	15
4.7 抽取规则	17
4.8 抽取词表	18
4.9 抽取短语实例	18
五 其他辅助功能	19
5.1 树图打印	19
5.2 文件管理	19
致谢	19

一 程序文件与界面

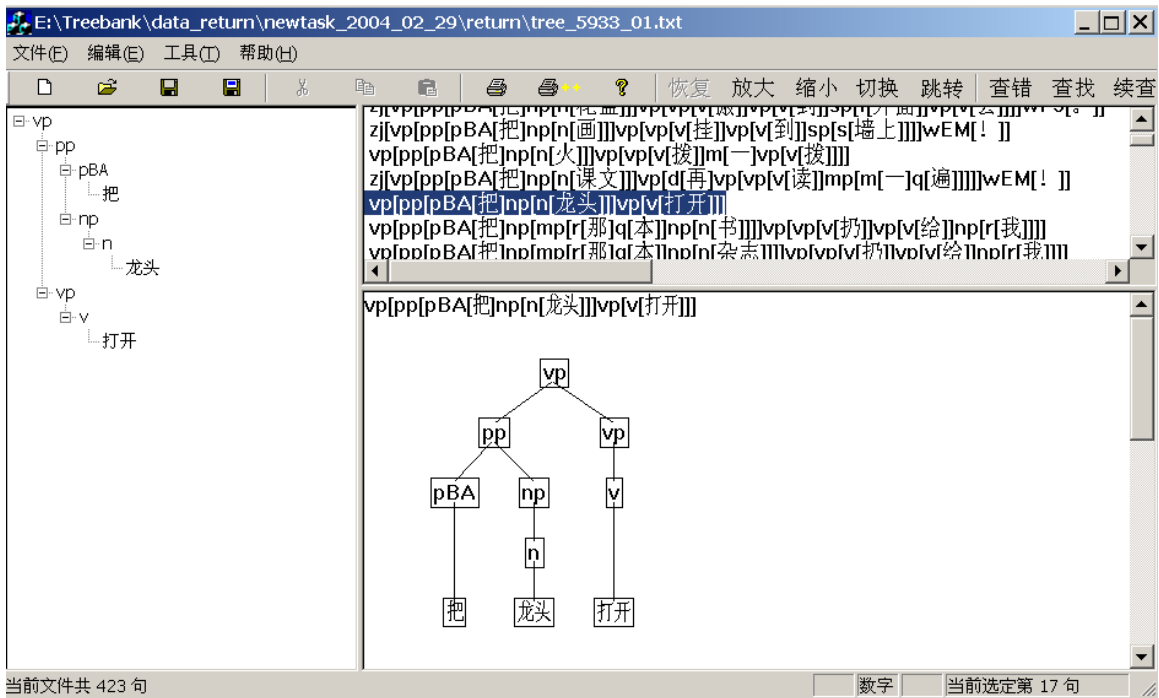
1.1 程序文件

本程序运行需要下面三个文件。

- 1) TreeEditor.exe, 可执行程序;
 - 2) syntag.txt, 定义树库中用到的各种标记, 帮助程序进行[标记的合法性检查](#);
 - 3) check.txt, 说明树库编辑过程中可能出现的结构错误, 用于进行[树结构的合法性检查](#)。
- 以上三个文件需要在同一个目录下(缺少后两个 txt 文件不影响程序的基本运行, 比如编辑操作, 但无法查错)。

1.2 程序界面

程序界面包含三大区域:



二 树库文件的格式

2.1 输入

1. 树库文件必须为文本文件（.txt, .TXT），文件格式为每行一个句子（如果一个句子被回车分成了两行，将作为两个句子看待）。**目前程序对树库文件的限制是不能超过10000个（行）句子，每句不超过5000个字节长度。**

2. 句子结构的表示必须遵循一定的格式。目前编辑工具支持如下三种格式：

格式1) 句子 = 节点 | 文字
节点 = 属性标记[节点] | 节点, 节点 | 叶子节点
叶子节点 = 属性标记[文字]
例句: s[np[猎人], vp[回来], u[了]]

格式2) 句子 = 节点 | 文字
节点 = 属性标记(节点) | 节点, 节点 | 叶子节点
叶子节点 = 属性标记(文字)
例句: s(dj(np(r(本文))vp(vp(v(讨论)u(了)ap(a(新)))np(n(一代)))np(np(ap(np(n(计算机))u(的))np(n(结构)))c(和)np(n(特点))))))

以上两种格式可以在标记和叶子节点之间任意插入空格，不影响显示，兄弟节点之间的逗号也可以省略，程序能自动判断。

格式3) 句子 = 节点 | 文字
节点 = [属性标记 节点] | 节点, 节点 | 叶子节点
叶子节点 = [属性标记 文字]
例句: [s [np 猎人] [vp [v 回来] [u 了]]]

对于上面这种格式，标记与叶子节点间一定要有空格分开（比如“np”和“猎人”之间）。（注：对这种格式，程序在内部处理时将转换成上面第一种格式）

上述三种格式中，“属性标记”前均可以再添加！（西文感叹号），这个符号的含义是**中心词标记**，它标记了它后面的成分在短语结构中是中心成分。比如：

(np (ap (a (新)) !np (n (家庭))))

在上面这个短语结构中，“!np (n (家庭))”是整个np的中心成分。

2.2 输出

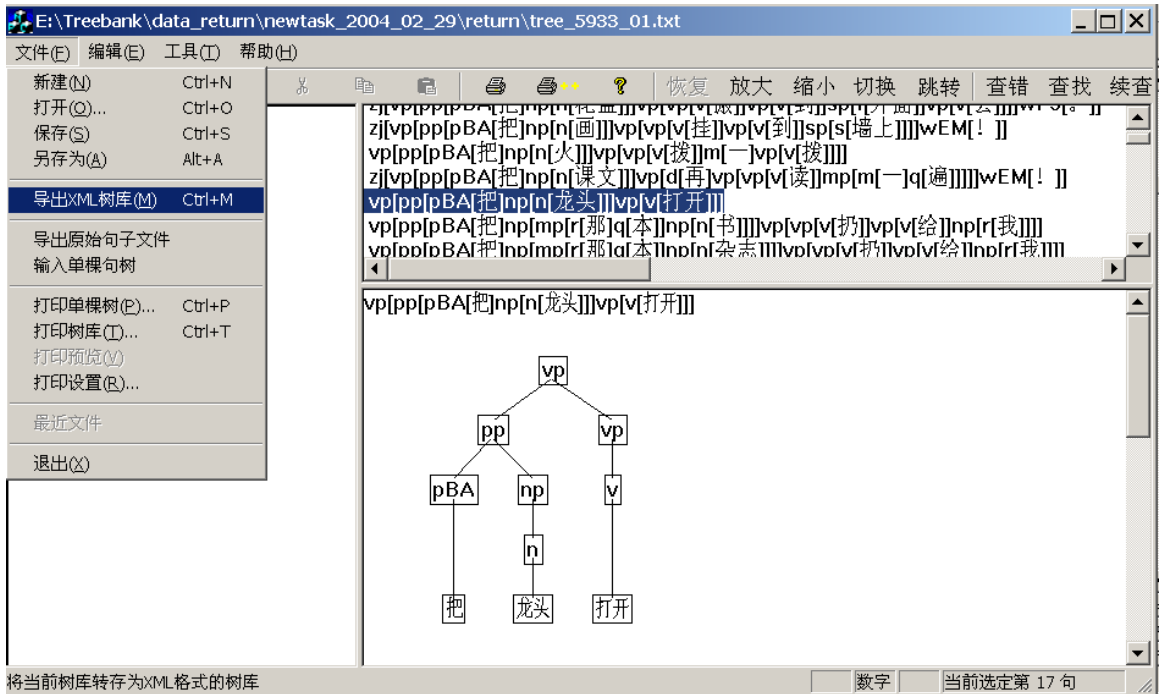
2.2.1 表 (list) 输出格式

1. 点击“保存图标”按钮或“文件”菜单项下“保存”命令（见2.2.2中图），可以保存对句子结构所做的修改。程序会对句子结构进行合法性检查，通过检查后才能顺利保存。树库中所允许

的句法标记集存放在一个文本文件（syntag.txt）中，程序读入其中的标记作为判断句子结构（标签）合法性的依据。点击“保存”，树库文件按2.1中所示“格式1”形式保存；点击“另存为”，树库文件按2.1中所示“格式2”形式保存（头、尾增加了一对圆括号）。

2. 如果做了编辑操作后没有保存就退出程序，会弹出对话框提示用户保存。

2.2.2 XML 输出格式



2.2.3 导出原始句子文件

点击 [文件] 菜单项下的“导出原始句子文件”选项，可以导出当前树库文件的原始句子形式。

三 句子结构树的显示与编辑

3.1 显示

打开文件后，所有句子按每句一行在右上角的窗口（“句子列表窗口”）中显示，任意选择其中一句，右下的窗口中显示该句子的图形树状结构，左边的窗口显示该句子的层次树状结构。图形树状结构便于操作者观察，快速找出句子结构的错误之处，层次结构便于编辑（节点拖动，属性修

改，以及添加节点等）。

3.1.1 树图的缩小与放大

在图形树状结构显示窗口中，如果图形超出了屏幕范围，可以通过拖动滚动条来查看其它部分，也可以点击工具栏上的“缩小”按钮来减小节点间的距离；如果节点距离太近以至出现节点交叠的情况，可以点击工具栏上的“放大”按钮来增加节点间距离。

3.1.2 句子列表显示模式切换

在句子列表窗口中，如果想查看不带句法结构标记的原始句子，可以点击工具栏上的“切换”按钮，来进行显示模式的切换。

3.2 编辑

3.2.1 节点拖动

将一个节点（包括它的所有子孙节点）拖动到其他位置，同时删除该节点原来所在位置。

拖动操作包括两种基本情况：

i) 设有非叶子节点 x ，对 x 执行拖动操作，是指将 x 从原位置移走，成为非叶子节点 y 的儿子节点。这种情况下，在拖动前要满足条件： x 不能是 y 的儿子节点。

这种拖动操作的影响对象是节点 x 下所有的节点，即整个一棵子树 x 。

执行上面这种拖动操作会有两种情况：

拖动后， x 作为 y 的第一个儿子节点，如果 x 所辖叶子节点先于 y 所辖叶子节点

拖动后， x 作为 y 的最后一个儿子节点，如果 y 所辖叶子节点先于 x 所辖叶子节点

ii) 设有叶子节点 x ，对 x 执行拖动操作，是指将 x 从原位置移走，与另一个叶子节点 y 合并为一个叶子节点。

执行这种拖动操作后，如果 x 的父亲节点没有其他子女，则自动删除 x 的父亲节点，并且按照这种原则依次向上删除 x 的祖先节点。

这种拖动操作实际上是为了实现叶子节点的合并，即解决分析中存在的分词错误问题（将不该分的词分开成两个词了）。

比如树中有两个叶子节点，分别是“计算”和“机”，如果这是分词错误，有必要把这两个叶子节点合并为一个，成为“计算机”。具体操作方法为：将叶子节点“机”拖到“计算”的父亲节点（记作 N_f ）上，两个叶子节点即合并为一个，成为 N_f 的儿子节点。

3.2.2 节点删除

叶子节点是不允许删除的。删除节点操作只针对非叶子节点。

设有x, y, z三个节点, x是y的父亲节点, y是z的父亲节点, 如果将z节点拖动到x节点下, 使得z成为x的儿子节点, 则y节点被删除。

拖动叶子节点可能造成叶子节点的祖先节点被删除, 可参见3.2.1的说明。

3.2.3 修改标签

将一个节点的标签 (label) 修改为另一个名称, 比如一个节点被分析为np结构, 但实际上是vp结构, 这时应将节点标签 np改为vp。

具体操作与在windows资源管理器 (或“我的电脑”) 环境下修改文件名完全一样, 即先用鼠标左键单击该节点, 当光标定位到该节点所在方框内部时, 即可对框内字符进行编辑操作。

3.2.4 插入节点

可以在一个非叶子节点x后插入一个兄弟节点y。具体操作方法如下:

在x节点上点击右键, 就会在x之后产生一个空节点y, 可以用跟“修改节点的标签值”同样的方式, 在y中输入适合的标签。

注意: 上述操作均不允许改变原句子中字符的序关系!!!

3.2.5 恢复

对于在树图操作窗口中所做的操作, 可以点击工具栏上的“**恢复**”按钮, 恢复上一步状态直至最初状态。其它窗口内容也会同步更新。

3.2.6 整句删除

在句子列表显示子窗口, 选定需要编辑的句子 (被选中的行以蓝色显示), 点击鼠标右键, 弹出菜单有两个选项, **delete**表示删除当前选定句子, **edit**表示对当前选定句子进行文本编辑操作。

选择delete, 当前被选定句子即从句子列表中删除, 原选定句子行成为一个空行。

注意: 删除是无法恢复的操作。

3.2.7 字符串编辑模式

在句子列表显示子窗口, 选定需要编辑的句子 (被选中的行以蓝色显示), 点击鼠标右键, 弹出菜单有两个选项, **delete**表示删除当前选定句子, **edit**表示对当前选定句子进行文本编辑操作。

选择edit, 弹出文本编辑框, 被选定句子及其句法结构均显示在文本框中, 可以手工加入标签 (np, vp, ...), 修改词语 (处理分词错误, 把应该分开而没有分开的词分开)。

编辑结束后, 按 [OK] 按钮返回, 所有子窗口内容随即更新。

注意: 字符串编辑模式下所做的编辑操作是无法恢复的。

四 辅助编辑功能

4.1 跳转

按下“**跳转**”按钮，可以在弹出的对话框中输入数字，程序会在句子列表框中定位到指定行。

4.2 查找

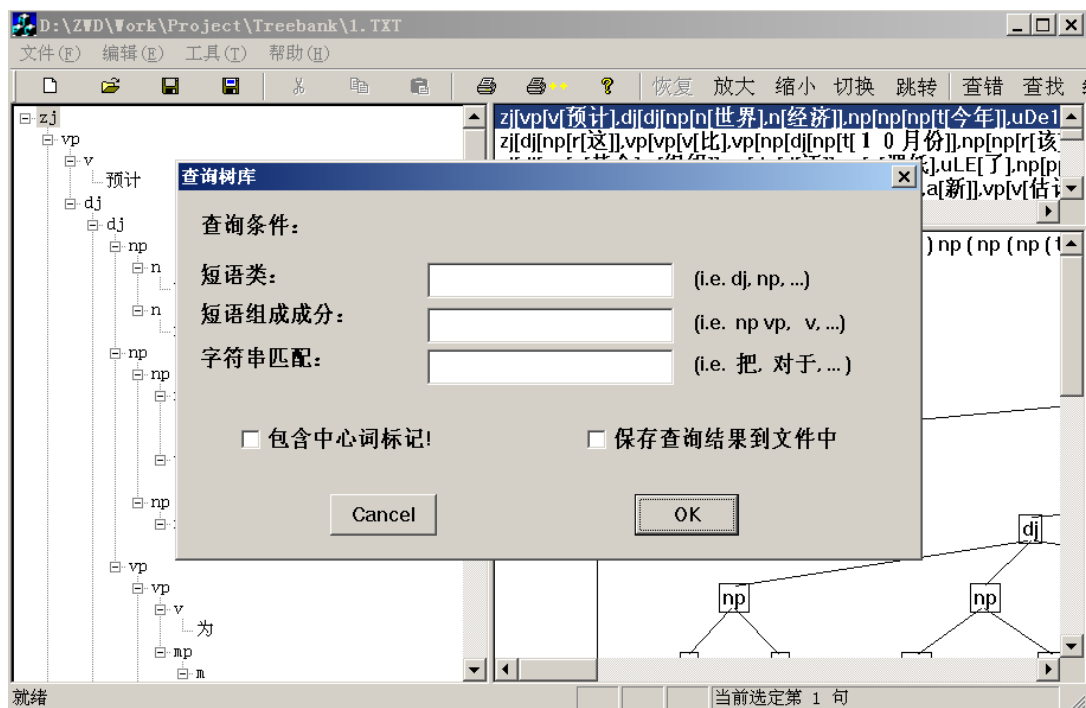
按下“**查找**”按钮，弹出对话框，用户可输入查找条件，确定后，程序会在句子列表框中定位到符合查找条件的句子所在的行。

查找条件有两种模式：

- (1) 树结构查找模式：这种情况下要求输入根节点和儿子节点作为查找条件
在这种查找模式下，还可以指定是否“包含中心词标记！”
缺省（default）的查找条件是“不包含中心词标记”，即在查找时程序将跳过树库中的!标记；如果用户选择“包含中心词标记”，则在查找时，将包含!号，进行匹配。

- (2) 字符串查找模式：这种情况下可输入任意字符串作为查询条件

每次只能按照一种模式查询。两种查找模式都可以选择将查询结果保存到用户指定的文件中。如果不指定保存文件，则程序弹出消息框，报告离当前行最近的符合查找条件的句子。



查找条件举例：

查询树库

查询条件：

短语类： np (i.e. dj, np, ...)

短语组成成分： np uDe1 np (i.e. np vp, v, ...)

字符串匹配： (i.e. 把, 对于, ...)

包含中心词标记! 保存查询结果到文件中

Cancel OK

上述查找条件下的查找目标为：（1）查np短语，该np短语由np + uDe1(的) + np组成
（2）此次查找不包含中心词标记！，结果也不输出到文件

注意，“查找”条件只能指定父亲节点和儿子节点（两层），不能嵌套更多层次，各儿子节点之间需有空格分隔。下面4.4“高级查找”功能支持层次嵌套的树结构查找。

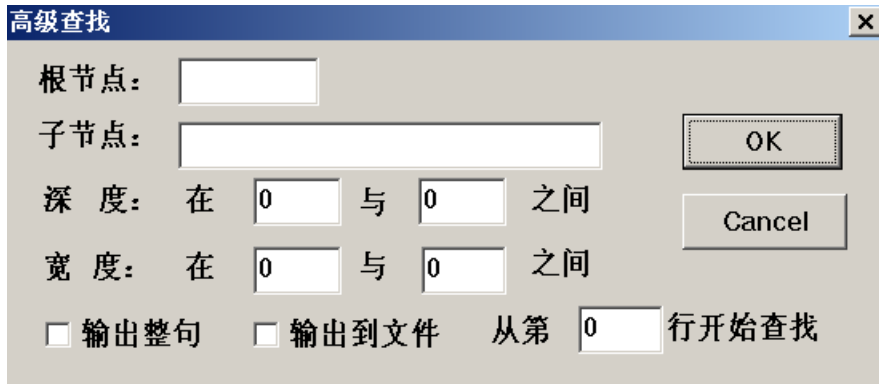
4.3 续查

查询条件不变，希望查找更多符合查询条件的句子，可按“续查”按钮（Find Next）。要改变查询条件，则按“查找”按钮。“查找”范围为整个文件，“续查”范围为从当前句开始到文件末尾。

4.4 高级查找

支持多层树结构的查找，用户可以指定树的根节点，儿子节点，孙子节点，树的宽度，深度等等作为查找条件。根节点的儿孙节点用“表”（list）形式给出，同层节点之间以逗号分隔。

下面是用户指定“高级查找”条件的对话框。



目前“高级查找”功能支持的组合查找条件包括：

- (1) 同时指定 根节点 + 子节点 （此时深度=宽度= 0，即深度和宽度不作为查找条件）
- (2) 同时指定 根节点 + 深度范围 （深度取值为大于0，小于100的整数）
- (3) 同时指定 根节点 + 宽度范围 （宽度取值为大于0，小于100的整数）

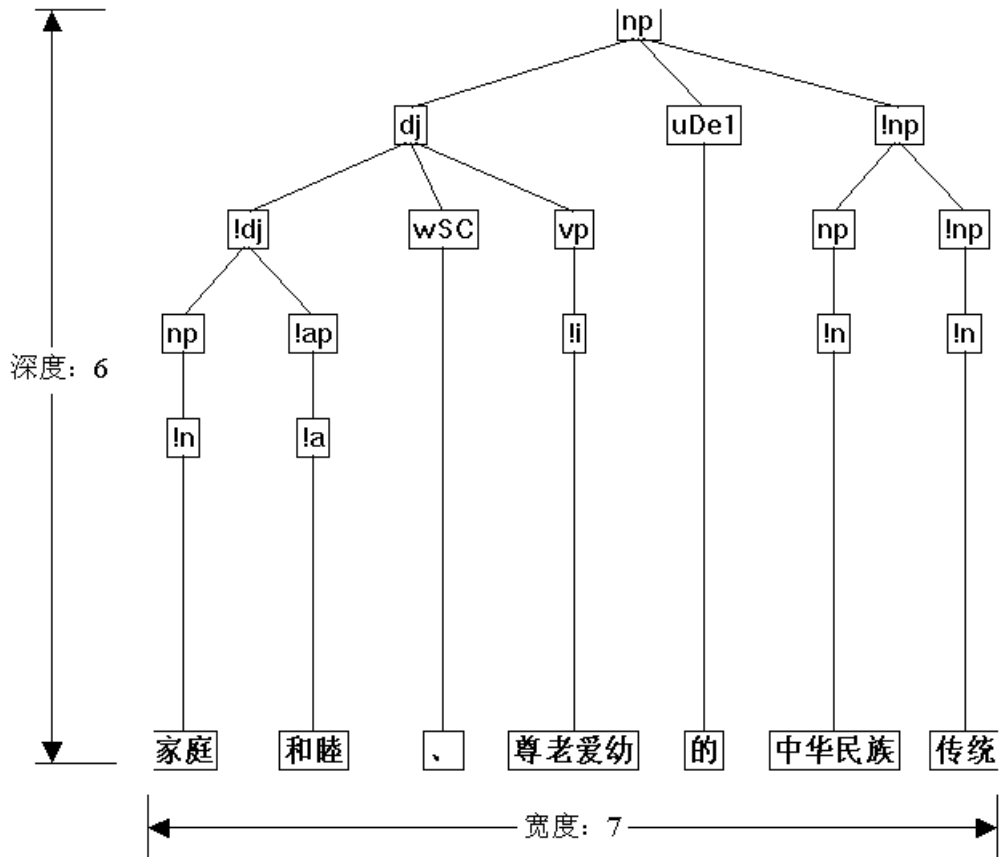
上述三种查找模式是排他的，并且是有先后顺序的，即如果指定了第一种查找模式，则第二、三种查找模式无效，依次类推。

用户可以选择将查找结果输出到文件，也可以不输出到文件，而是定位到符合查找条件的句子所在的行；

用户可以指定查找的起始行；

如果用户选择将查找结果输出到文件，在上述第（1）种查找模式下，用户可以选择输出结果为命中的整句树结构，或者是完全匹配的子树（短语）；在第（2），（3）查找模式下，输出结果均为完全匹配的子树（短语），而不是输出整句。也就是说，在第一种查找模式下，“输出整句”选项是有效的，但在第二、三种模式下，“输出整句”选项不起作用。

树的深度是指从根节点到叶子节点的最大层数。树的宽度是指一棵树的叶子节点的个数。树的深度和宽度可以反映一个句法结构的复杂程度。比如下面这个np树的深度为6，宽度为7。



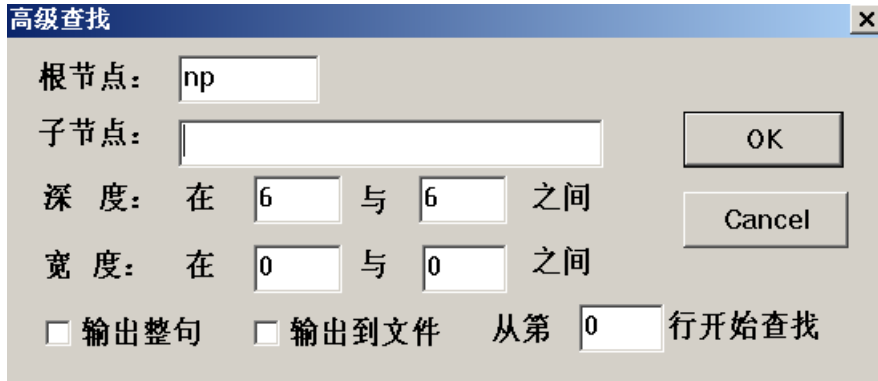
注意：高级查找时对中心词标记是敏感的，例如“!np”和“np”将被看作是不同的节点。

下面是查找条件设置示例：

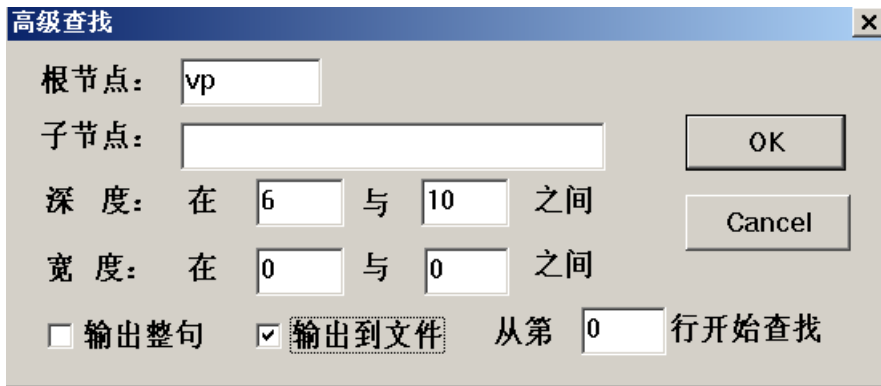
(1) 比如在根节点栏填 np，在儿子节点栏填：np[n],np[n,n]

表示： 查找一个np结构，该np由两个np构成，其中第一个np由一个n组成，第二个np由两个n组成。图示如下：

(2) 在下面查找示例中，用户可以在树库中找到深度为6的np子树。



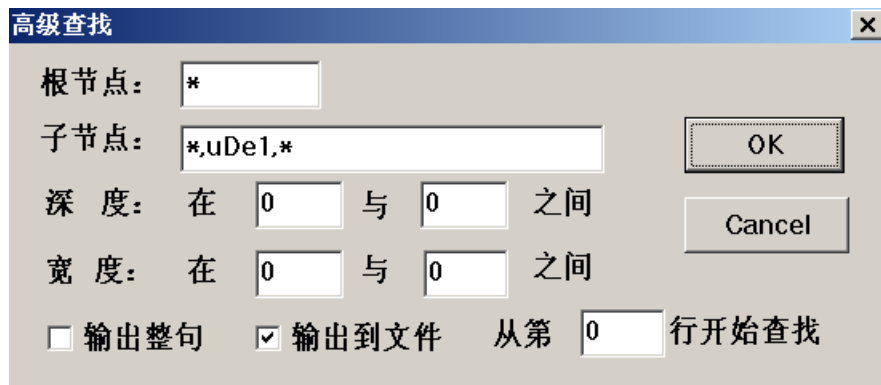
(3) 在下面的查找中，用户可以在树库中查找深度从6到10之间的vp短语，并将结果保存到文件。



特别提示：

[高级查找]支持通配符（用“*”表示）查找。根节点和子节点中均可以包含*。比如：

(4) 在下面的查找中，用户可以在树库中查找所有的中间包含“的”字的三分支结构，并将查找结果输出到文件。



(5) *可以跟其他字母配合起来使用，比如“!*”表示跟“!”开头的字符串匹配；“*j”表示跟“j”字母结尾的字符串匹配；“!*j”表示跟“!”开头，“j”结尾的字符串匹配。在下面的查询中，用户可以在树库中查找到所有中间包含“的”字的三分支结构，并且，“的”

字左边的节点是以“j”结尾的。

高级查找

根节点: *

子节点: *j,uDe1,*

深度: 在 0 与 0 之间

宽度: 在 0 与 0 之间

输出整句 输出到文件 从第 0 行开始查找

OK Cancel

(6) 下面这个示例查找“的(uDe1)”字结构作中心语的短语结构，即根节点为“!*”，指跟所有以!开头的节点匹配。

高级查找

根节点: !*

子节点: *,uDe1

深度: 在 0 与 0 之间

宽度: 在 0 与 0 之间

输出整句 输出到文件 从第 0 行开始查找

OK Cancel

特别提示:

[高级查找] 支持逻辑非(用“-”号表示)查找。比如“-np”表示“不是np”的短语。注意：-只能出现在一个节点标签值的开头。像“n-p”“!-np”这样的标签是非法的。

(7) 下面这个示例查找根节点不是ap的“的(uDe1)”字短语。

高级查找

根节点: -ap

子节点: *,uDe1

深度: 在 0 与 0 之间

宽度: 在 0 与 0 之间

输出整句 输出到文件 从第 0 行开始查找

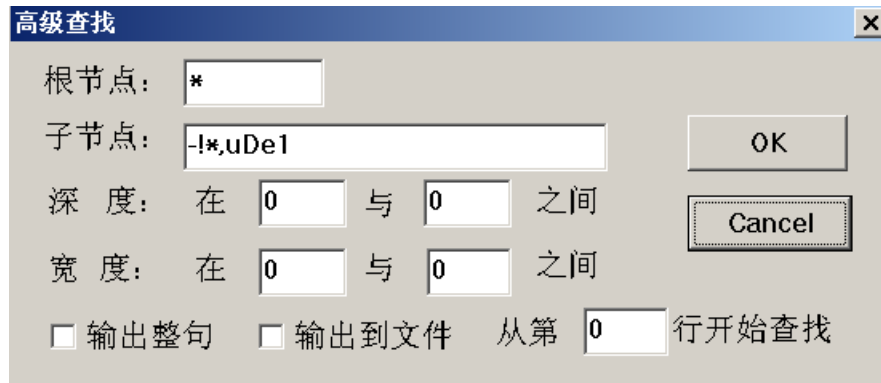
OK Cancel

(8) 下面这个示例查找不是“的(uDe1)字短语的np短语。



高级查找对话框的截图。根节点输入框为“np”，子节点输入框为“*,-uDe1”。深度和宽度均设置为在0与0之间。底部有“输出整句”、“输出到文件”复选框以及“从第0行开始查找”的输入框。右侧有“OK”和“Cancel”按钮。

(9) 下面这个示例查找“的(uDe1)”的兄弟节点不含!标记的子树。在查找时，“-!*”表示一个节点不是以!开头的。(注意，在“替换子树”中，“-!*”表示把节点标签中开头的!删掉，见下文4.5中的说明)



高级查找对话框的截图。根节点输入框为“*”，子节点输入框为“-!*uDe1”。深度和宽度均设置为在0与0之间。底部有“输出整句”、“输出到文件”复选框以及“从第0行开始查找”的输入框。右侧有“OK”和“Cancel”按钮。

注意：目前一个节点中仅可以包含一个*。像“*p*”“***”“!*j*”这样的串都是非法的。

如果选择“输出到文件”，按下上面对话框的“OK”按钮后，用户可以进一步指定查找结果的保存位置和文件名。

4.5 子树替换

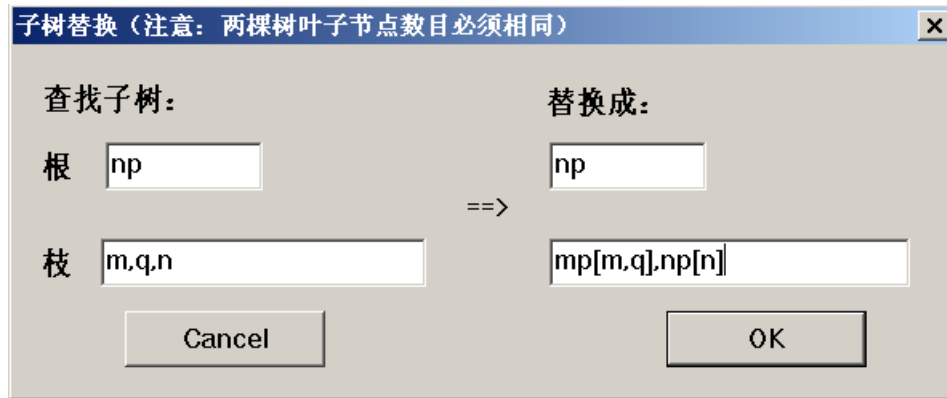
支持树结构的替换操作，程序首先根据用户指定的树结构条件，在树库中查找符合用户指定条件的句树，并将相应内容替换为用户指定的目标树结构。查找方式与“高级查找”(4.4)相同。

注意：替换时对中心词标记是敏感的，例如“!np”和“np”将被看作是不同的节点。

替换后的结果以文件形式保存。用户可以指定保存位置和文件名。

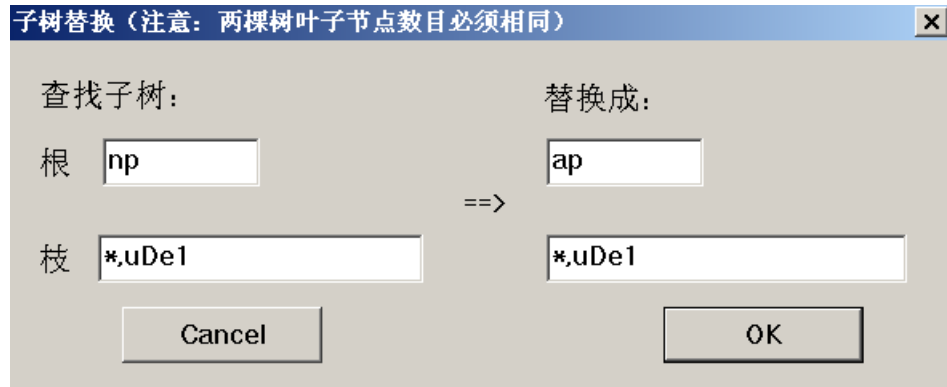
下面是示例：

(1) 在下面的替换中，np原本是三支结构(两层)，由m+q+n组成；替换后成为二支结构(三层)。由mp+np组成，其中mp又由m+q组成，np由n组成。



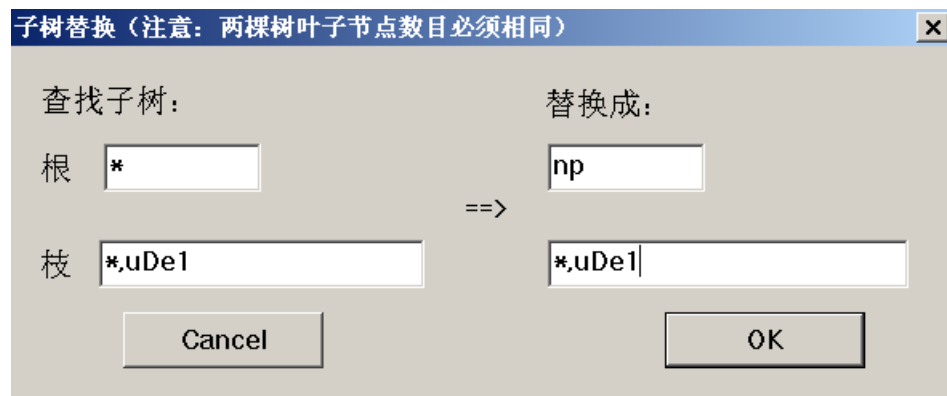
注意：替换前后叶子节点数目不能改变（比如上例中都是三个：m,q,n）。当然，节点的标签值是可以更改的（比如原来的m可以改为任何一个其他的合法的词性标记）。

(2) 在下面的替换中，np子树（二分支，其中后一个儿子节点为uDe1）被替换为ap子树（二分支，其中前一个儿子节点继承替换前np的第一个儿子节点，后一个儿子节点仍为uDe1）。

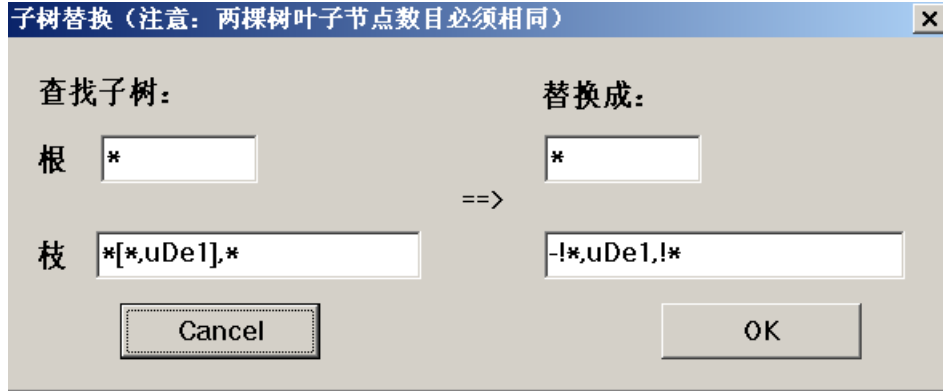


子树替换操作允许在树结构中使用 * 作为通配符，* 代表一个节点，可以跟任何一个标记匹配。

(3) 在下面这个替换中，任意一个包含“的”（uDe1）的二分支子树，都将被替换为np子树，np的第一个节点从替换前子树的第一个儿子节点继承，np的第二个节点仍为uDe1。

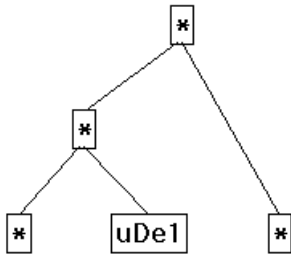


(4) 在下面这个替换中，任意一个包含“的” (uDe1) 的三层二分支子树，将被替换为一个两层三分支子树。并且，替换之前，uDe1的兄弟节点如果含有！标记，替换后，将不再含有！标记；此外，替换后，uDe1的后一个兄弟节点将含有！标记，无论在替换前它是否含有！标记。

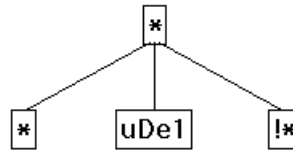


替换前后的树形图如下：

替换前：



替换后：



关于“子树替换”操作，需要注意：

(1) “查找子树”部分，跟“高级查找”中用户指定待查找子树的方式完全一样。可以包含*， *j， n*， !*p， !n*， -!*， -np， …… 等等字符串形式，其中通配符 * 可以跟其他任意字母配合起来使用，非号“-”只要出现在开头，也可以跟其他字母配合使用。

(2) “替换”部分，通配符 * 只有三种形式（相应的，非号 - 只有其中的一种形式）：

- * 表示在替换时完全继承 [查找子树] 部分对应的节点标签值；
 - !* 表示在替换后，在源节点标签前加上！标记，如果原来没有！标记的话；
 - !* 表示在替换后，将源节点标签前的！标记删除，如果原来有！标记的话；
- （其中 - 是西文减号）

“替换”部分不允许出现 *j， n*， !*p， !n*， …… 等等 * 跟字母连用的形式；也不允许 -np， -!*np 这样的形式。

无论是查找，还是替换，节点中只能包含一个 *。不允许含多个 * 的形式。“-”号只能出现在开头。

4.6 查错

程序目前能检查出来的“错误”主要是不符合规范的“硬伤”（不涉及到语言学知识），另外程序也对一些可能存在的“软伤”（涉及到语言学知识），猜测性地做出报告（在树库加工中操作

人员可能判断有误，或者是误操作，造成一些错误的组合模式，程序尝试发现这类错误）。目前程序定义的错误包括以下类型：

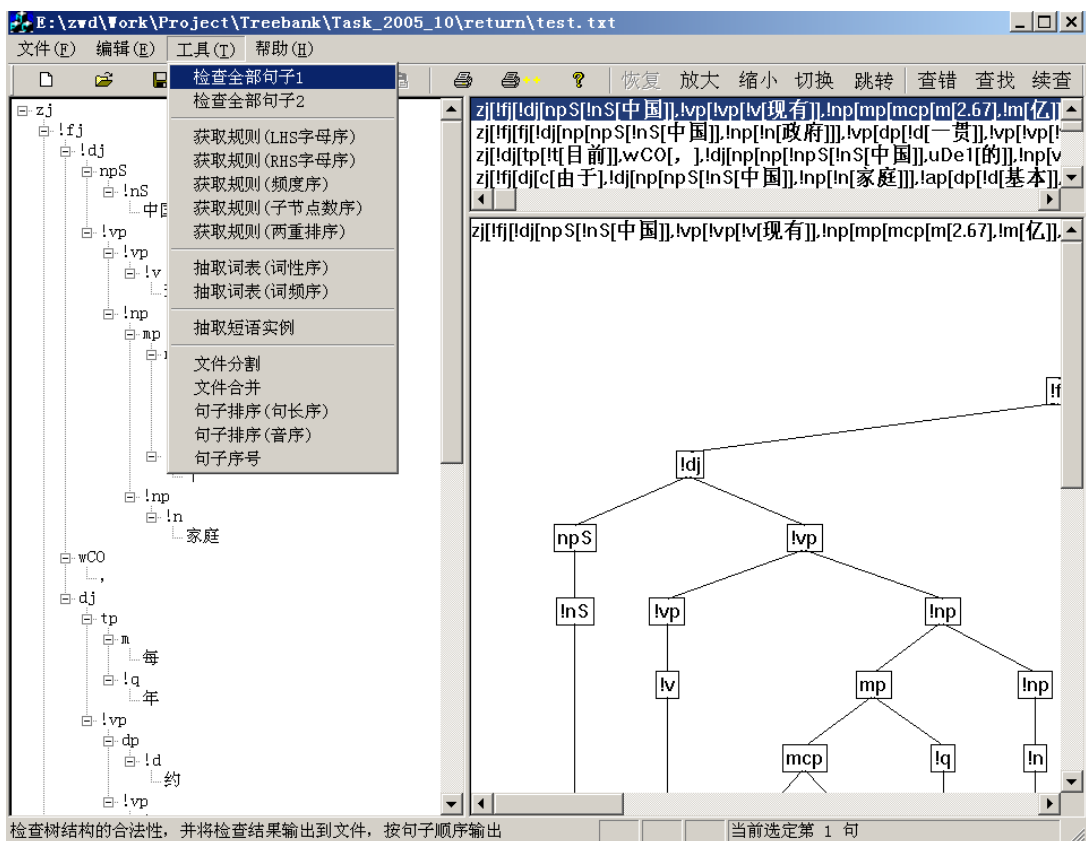
- (1) 子节点数超过5个，并且节点中不含标点；
- (2) 把汉字标记为标点；
- (3) 后置成分充当短语的开头；
- (4) 形如np->np，或 dj->vp这样的单分支节点错误；
- (5) 形如q->q n这样的由词性标记充当短语类的错误；
- (6) check.txt文件中定义的各种错误概率较高的组合类型。

注意：上述第六类“错误”有可能并不是错误，但树库加工人员应该谨慎对待程序报告的每一个错误提示信息，进行认真核查。

按下“**查错**”按钮，程序即对当前打开的树库文件逐句进行检查，在句子列表框中定位到包含错误的句子所在的行。定位后，用户可以用鼠标点击该句子，程序即会显示当前可能有错误的句子，等待用户进行编辑操作，如果程序报告的错误属实，用户修改后点击“查错”，该句即可通过检查；如果程序报告的错误属于误报，则用户可选中下一个句子，然后点击“查错”，即跳过刚才报错的句子，继续进行检查。在这种查错模式下，每次查错范围：以当前选定句子为起始，最后一句为结束。查到第一个错误即停止。

按下[工具]菜单项下的[**检查全部句子1**]项，程序对当前树库文件中的全部句子进行检查，并将检查结果输出到文件中保存。检查结果的文件名命名原则是：如果当前树库文件名为 MyTree.txt，则检查结果文件名为 MyTree_err1.txt，与树库文件在同一目录下。在这种查错模式下，无论当前用户选定哪一个句子，查错范围都是整个文件。检查结果按照句子原始顺序排序，即第i句中包含的错误信息先于第i+1句的错误信息。

按下[工具]菜单项下的[**检查全部句子2**]项，程序对当前树库文件中的全部句子进行检查，并将检查结果输出到文件中保存。检查结果的文件名命名原则是：如果当前树库文件名为 MyTree.txt，则检查结果文件名为 MyTree_err2.txt，与树库文件在同一目录下。在这种查错模式下，无论当前用户选定哪一个句子，查错范围都是整个文件。检查结果按照所查到的错误信息的先后顺序输出，每条错误信息后面记录包含了这类错误的句子的序号。



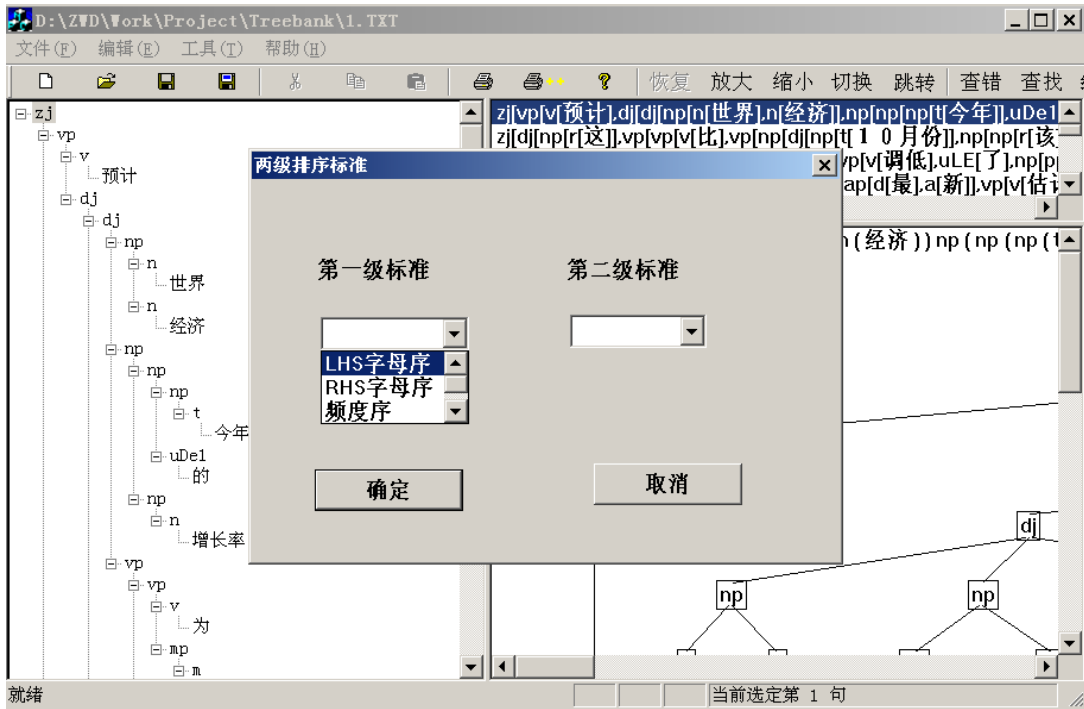
4.7 抽取规则

按下[工具]菜单项下的[获取规则 (...)]等项目(参见4.4中图示),程序可以对当前已打开的树库文件,或者在当前无打开的树库文件时,打开一个新的树库文件(这时对文件中的句子数量无限制),进行分析,获取其中的产生式规则(可以由用户指定不同的排序方式)。对于前者,用户可以在文件保存对话框中指定结果文件的保存位置和文件名;对于后一种打开文件的方式,程序自动以所打开的树库文件名后接_rule作为结果文件的文件名,保存在跟树库文件相同的目录下。

进一步分析规则文件,可以帮助用户发现树库中可能存在的错误,从而返回树库编辑环境进行编辑操作(自动抽取的规则中,低频规则有更大的可能意味着句法结构标注错误)。

点击“获取规则(两重排序)”,会弹出对话框,由用户指定规则输出到文件时的排序标准。用户可以指定两级排序标准,也可以只指定一个排序标准。如果不指定排序标准,则不进行排序,直接将树库中包含的所有规则,按照原句顺序输出到文件。

从功能上说,“获取规则(两重排序)”菜单项的功能覆盖了其他“获取规则…”菜单项的功能。但考虑到使用方便,当用户不需要多重排序标准对规则进行排序时,可以直接点击其他“获取规则…”菜单项,而不需要选择“获取规则(两重排序)”菜单项。点击“获取规则(两重排序)”菜单项后弹出的对话框如下图所示。用户可以根据需要在下拉列表框中选择排序标准。



4.8 抽取词表

按下[工具]菜单项下的[抽取词表 (...)]等项目，程序对当前树库文件进行分析，获取其中带词性标记的词表（可以由用户指定不同的排序方式）。用户可以在文件保存对话框中指定结果文件的保存位置和文件名。

4.9 抽取短语实例

按下[工具]菜单项下的[抽取短语实例]，程序对当前树库文件进行分析，获取用户指定短语类型的短语实例，包括从树库中抽取 根节点=XP 的最大XP短语和所有XP短语两种选择。这里最大XP短语是指在一个句子中，不被XP覆盖的XP短语。

提示：以上[辅助编辑]功能也可以作为利用树库开展语言研究和教学的[辅助工具]。比如[查找][高级查找]可以帮助我们从中提取各种类型的短语结构进行针对性的研究和教学；[子树替换]功能可以帮助我们批量修改树库中的某种结构模式，从而有可能对比修改前后（即对同一个语言片断，有不同的句法结构分析），对概率句法分析器会造成什么影响；[抽取规则][抽取词表]则可以帮助我们从宏观整体的角度了解一个语言的各类词语的分布情况和结构组合类型及其分布情况；等等。


五 其他辅助功能

5.1 树图打印

1) 打印单棵句树

点击“打印机图标”按钮 ，打印当前句子树图；

2) 打印整个树库文件

点击带有两个加号（“++”）的“打印机图标”按钮 ，打印当前树库文件中所有句子的树图。

5.2 文件管理

为方便文件操作（校对），在“工具”菜单项下提供了“文件分割”“文件合并”“句子排序”“句首前添加序号”等功能。

“文件分割”可以将一个大文件均分成若干小文件（以行/句数均分）；

“文件合并”可以将一个目录下的多个树库文件合并为一个文件，文件名为：目录名.txt。合并后的文件存放在原树库文件的上层目录下。

“句子排序”提供按“音序”和“句长序”排序输出树库句子的功能；

“句首前添加序号”即在每个句子所在的行首添加一个序号，从1开始顺序编号。这项功能对应的菜单项为“句子序号”。

具体菜单可以参看4.4中的图示。

—— 正文结束 ——

致谢

北大计算语言所常宝宝博士提供了字符串与树结构之间的转换等一批函数；2001级硕士研究生吴拥华同学实现了树图显示以及树节点编辑的原型功能。感谢他们为设计和实现TreeEditor提供的帮助。